IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In Re Application of:	Date: April 2, 2007
Robbert C. VAN DER LINDEN, et al.	Confirmation No. 3744
Serial No: 10/648,752	Group Art Unit: 2162
Filed: August 25, 2003	Examiner: Giovanna B. COLAN

For: METHOD AND SYSTEM FOR QUERYING STRUCTURED DOCUMENTS STORED IN THEIR NATIVE FORMAT IN A DATABASE

Mail Stop Appeal Brief – Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

SUBSTITUTE APPEAL BRIEF

Dear Sir or Madam:

In response to the Office Communication mailed March 28, 2007, Appellant submits this Substitute Appeal Brief pursuant to 37 C.F.R. § 41.37.

I. REAL PARTY IN INTEREST

The real party in interest is International Business Machines Corp. of Armonk, New York by virtue of an assignment from the inventor(s) recorded in the U.S. Patent and Trademark Office on August 25, 2003, at Reel No. 014443, Frame No. 0420.

II. RELATED APPEALS AND INTERFERENCES

There are no appeals, interferences, or judicial proceedings known to Appellant, the Appellant's legal representative, or Assignee, which may be related to, directly affect, be directly

affected by, or have a bearing on the decision by the Board of Patent Appeals and Interferences in the pending appeal.

III. STATUS OF CLAIMS

Claims 1, 3-11, 13-21, and 23-36 have been rejected. Appeal is taken from the rejection of claims 1, 3-11, 13-21, and 23-36.

IV. STATUS OF AMENDMENTS

No amendments were filed subsequent to the final Office action dated August 22, 2006.

V. SUMMARY OF CLAIMED SUBJECT MATTER

The present invention is directed to "method and system for querying structured documents stored in their native format in a database system" (pg. 4, lns. 18-19). "Through aspects of the present invention, a structured document is parsed and a plurality of nodes is generated to form a hierarchical node tree representing the structured document. The plurality of nodes is stored in one or more records. Each node that has children includes a plurality of child pointers. Stored in each child pointer is a hint related to the child node to which the child pointer points. . . . By storing the hint in the child pointer, a database management system (DBMS) navigating the node tree during query evaluation follows the pointers that contain a hint that matches a query and can skip over those that contain non-matching hint. Accordingly, query processing is more efficient" (pg. 16, ln. 19 to pg. 17, ln. 5).

Independent claim 1 recites a method for querying a structured document (202) stored in its native format in a database, wherein the structured document (202) comprises a plurality of

nodes (508, 800) that form a hierarchical node tree (208). *See, e.g.*, pg. 7, lns. 3-6; pg. 14, lns. 12-21; figs. 2, 4, 7A, and 8-9. The method includes providing at least one child pointer (510) within at least one of the plurality of nodes (508, 800) in the hierarchical node tree (208), wherein the at least one child pointer (510) points to a corresponding child node (508) in the hierarchical node tree (208) (902). *See, e.g.*, pg. 14, lns. 13-15; figs. 2, 4, 7A, and 8-9. The method also includes storing a hint (804b) within the at least one child pointer (510), the hint (804b) being related to the corresponding child node (508), wherein the at least one child pointer (510) further comprises a node slot number (804a) of the corresponding child node (508) (904). *See, e.g.*, pg. 14, lns. 15-16; figs. 4, 7A, and 8-9. The method further includes utilizing the hint (804b) to determine whether to navigate to the corresponding child node (508) during query evaluation (906). *See, e.g.*, pg. 14, lns. 16-18; figs. 4, 7A, and 8-9.

Independent claim 11 recites a computer readable medium containing a computer program for querying a structured document (202) stored in its native format in a database, wherein the structured document (202) comprises a plurality of nodes (508, 800) that form a hierarchical node tree (208). *See, e.g.*, pg. 7, lns. 3-6; pg. 14, lns. 12-21; figs. 2, 4, 7A, and 8-9. The computer program includes instructions for providing at least one child pointer (510) within at least one of the plurality of nodes (508, 800) in the hierarchical node tree (208), wherein the at least one child pointer (510) points to a corresponding child node (508) in the hierarchical node tree (208) (902). *See, e.g.*, pg. 14, lns. 13-15; figs. 2, 4, 7A, and 8-9. The computer program also includes instructions for storing a hint (804b) within the at least one child pointer (510), the hint (804b) being related to the corresponding child node (508), wherein the at least one child pointer (510) further comprises a node slot number (804a) of the corresponding child node (508) (904). *See, e.g.*, pg. 14, lns. 15-16; figs. 4, 7A, and 8-9. The computer program further includes

instructions for utilizing the hint (804b) to determine whether to navigate to the corresponding child node (508) during query evaluation (906). *See, e.g.*, pg. 14, lns. 16-18; figs. 4, 7A, and 8-9.

Independent claim 21 recites a system for querying a structured document (202) stored in its native format in a database, wherein the structured document (202) comprises a plurality of nodes (508, 800) that form a hierarchical node tree (208). *See, e.g.*, pg. 7, lns. 3-6; pg. 14, lns. 12-21; figs. 2, 4, 7A, and 8-9. The system includes a computer system (104) coupled to at least one data storage device (106). *See, e.g.*, pg. 5, ln. 15 to pg. 6, ln. 5; fig. 1. The system also includes a database management system (105) in the computer system (104). *See, e.g.*, pg. 5, ln. 22 to pg. 6, ln. 5; fig. 1. The system further includes a storage mechanism (200) in the database management system (105) for providing at least one child pointer (510) within at least one of the plurality of nodes (508, 800) in the hierarchical node tree (208), wherein the at least one child pointer (510) points to a corresponding child node (508) in the hierarchical node tree (208), and storing a hint (804b) within the at least one child pointer (510), the hint (804b) being related to the corresponding child node (508), wherein the at least one child pointer (510) further comprises a node slot number (804a) of the corresponding child node (508), wherein the database management system (105) utilizes the hint (804a) to determine whether to navigate to the corresponding child node (508) during query evaluation. *See, e.g.*, pg. 14, lns. 13-18; figs. 2, 4, 7A, and 8-9.

Dependent claim 31 depends from independent claim 1 and recites wherein each of the plurality of nodes (508, 800) in the hierarchical node tree (208) specifies a type of node, one or more nodes in the hierarchical node tree (208) being of a text type and one or more other nodes

in the hierarchical tree (208) being of a non-text type. *See, e.g.*, pg. 12, lns. 19-21; figs. 2, 4, and 8.

Dependent claim 32 depends from independent claim 11 and recites wherein each of the plurality of nodes (508, 800) in the hierarchical node tree (208) specifies a type of node, one or more nodes in the hierarchical node tree (208) being of a text type and one or more other nodes in the hierarchical tree (208) being of a non text type. *See, e.g.*, pg. 12, lns. 19-21; figs. 2, 4, and 8.

Dependent claim 33 depends from independent claim 21 and recites wherein each of the plurality of nodes (508, 800) in the hierarchical node tree (208) specifies a type of node, one or more nodes in the hierarchical node tree (208) being of a text type and one or more other nodes in the hierarchical tree (208) being of a non-text type. *See, e.g.*, pg. 12, lns. 19-21; figs. 2, 4, and 8.

Dependent claim 34 depends from independent claim 1 and recites wherein the at least one node (508, 800) in the hierarchical node tree (208) further includes at least one other child pointer (510, 511), the at least one other child pointer (510, 511) pointing to itself or to an in-lined character array. *See, e.g.*, pg. 8, lns. 9-17; pg. 16, lns. 6-17; figs. 2, 4, 7A, 7B, 7C, and 8.

Dependent claim 35 depends from independent claim 11 and recites wherein the at least one node (508, 800) in the hierarchical node tree (208) further includes at least one other child pointer (510, 511), the at least one other child pointer (510, 511) pointing to itself or to an in-lined character array. *See, e.g.*, pg. 8, lns. 9-17; pg. 16, lns. 6-17; figs. 2, 4, 7A, 7B, 7C, and 8.

Dependent claim 36 depends from independent claim 21 and recites wherein the at least one node (508, 800) in the hierarchical node tree (208) further includes at least one other child pointer (510, 511), the at least one other child pointer (510, 511) pointing to itself or to an in-lined character array. *See, e.g.*, pg. 8, lns. 9-17; pg. 16, lns. 6-17; figs. 2, 4, 7A, 7B, 7C, and 8.

VI. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

1. Appellant requests review as to claims 1, 3-11, 13-21, and 23-36, and their rejection under 35 U.S.C. § 112, first paragraph, as failing to comply with the written description requirement.

2. Appellant requests review as to claims 1, 3-5, 9, 11, 13-15, 19, 21, 23-25, 29, and 31-36, and their rejection under 35 U.S.C. § 102(e) as being anticipated by U.S. Patent No. 6,853,992 to Igata (hereinafter "Igata").

3. Appellant requests review as to claims 6, 10, 16, 20, 26, and 30, and their rejection under 35 U.S.C. § 103(a) as being unpatentable over Igata, in view of U.S. Patent No. 6,836,778 to Manikutty et al. (hereinafter "Manikutty").

4. Appellant requests review as to claims 7-8, 17-18, and 27-28, and their rejection under 35 U.S.C. § 103(a) as being unpatentable over Igata, in view of U.S. Patent Application Publication No. 2004/0243553 to Bailey (hereinafter "Bailey").

VII. ARGUMENTS

1. Claims 1, 11, and 21 Satisfy the Written Description Requirement

Claim 1 recites a method for querying a structured document stored in its native format in a database, wherein the structured document comprises a plurality of nodes that form a hierarchical node tree. The method includes providing at least one child pointer within at least one of the plurality of nodes in the hierarchical node tree, wherein the at least one child pointer points to a corresponding child node in the hierarchical node tree, storing a hint within the at least one child pointer, the hint being related to the corresponding child node, wherein the at least one child pointer further comprises a node slot number of the corresponding child node, and utilizing the hint to determine whether to navigate to the corresponding child node during query evaluation.

In the final Office action, the Examiner states:

> Claim 1, 3-11, 13-21, and 23-36 are rejected under 35 U.S.C. § 112, first paragraph, as failing to comply with the written description requirement. The claim(s) contains subject matter which was not described in the specification in such a way as to reasonably convey to one skilled in the relevant art that the inventor(s), at the time the application was filed, had possession of the claimed invention.
>
> Regarding Claims 1, 11, and 21, the term "within" is not clearly defined by the claim language. The specification does not provide support for the limitation including one child pointer "within" at least one of the plurality of nodes. Examiner is unclear about how a pointer, which is a link that indicates as to where to navigate and/or search, can contain a hint within itself, and not in the tree node. In addition, the specification, specifically paragraph [0055] and Fig. 8, does not provide the details as to how the hint is within the pointer and how the pointer is within the node.

(August 22, 2006 final Office action, pgs. 2-3).

Prior to the Amendment of May 25, 2006, claim 1 recited "providing at least one child pointer in at least one node of the plurality of nodes" (emphasis added). However, in light of the Examiner's comments in the February 28, 2006 Office action and after speaking with the Examiner in an interview on May 16, 2006, Appellant amended claim 1 to recite "providing at least one child pointer within at least one of the plurality of nodes in the hierarchical node tree" (emphasis added), to make it clear that the "at least one child pointer" is contained inside the structure of a node, as clearly illustrated in Figures 4, 7A, 7B, 7C, and 8 and described in the corresponding description in the Specification.

Appellant respectfully submits that based on the illustrations in Figures 4, 7A, 7B, 7C, and 8, and the corresponding description in the Specification, one of ordinary skill in the art will readily understand that a node in the present application is a collection of data that is treated as a single unit, which is evidenced by, for example, the illustration of a box 508 in Figure 4 to represent a single node. In addition, as shown in Figures 4, 7A, 7B, 7C, and 8, and described in the corresponding description in the Specification, included within a node, which is illustrated as box 508 in Figure 4, is at least one pointer, which is illustrated as a smaller box 510 within box 508 in Figure 4.

With respect to the element "storing a hint within the at least one child pointer" recited in claim 1 (emphasis added), Appellant respectfully notes that under MPEP § 2111:

> An applicant is entitled to be his or her own lexicographer and may rebut the presumption that claim terms are to be given their ordinary and customary meaning by clearly setting forth a definition of the term that is different from its ordinary and customary meaning(s). See *In re Paulsen*, 30 F.3d 1475, 1480, 31 USPQ2d 1671, 1674 (Fed. Cir. 1994) (inventor may define specific terms used to describe invention, but must do so "with reasonable clarity, deliberateness, and precision" and, if done, must "'set out his uncommon definition in some manner within the patent disclosure' so as to give one of ordinary skill in the art notice of

> the change" in meaning) (quoting *Intellicall, Inc. v. Phonometrics, Inc.*, 952 F.2d 1384, 1387-88, 21 USPQ2d 1383, 1386 (Fed. Cir. 1992)).

(M.P.E.P. § 2111.01, 8th ed., 5th rev.).

As clearly shown in Figure 8 and described on pages 13-14 of the Specification, a pointer in the present application is more than just a link to another node, as asserted by the Examiner. Based on the illustration in Figure 8 and the corresponding description in the Specification, one of ordinary skill in the art will readily understand that a pointer in the present application is also a collection of data that is treated as a single unit, similar to a node in the present application. As defined in claim 1 and shown in Figure 8, data that is contained inside a pointer includes a hint and a node slot number.

Further, Appellant respectfully submits that one of ordinary skill in the art does not require a detail explanation as to how the hint is within the pointer and how the pointer is within the node, as asserted by the Examiner. One of ordinary skill in the art is aware that the nodes and pointers in the present application can be, for instance, objects instantiated from user-defined classes. As an example, a user can define a class named “node” that includes various data members and another class named “pointer” that also includes various data members. Each data member can be of a standard data type, such as an integer, a long integer, a character, a character string, and so forth, or be of a user-defined data type, such as another class.

Hence, the “pointer” class can be defined to include a data member named “hint” that is, for instance, of a character string data type, and another data member named “node slot number” that is, for instance, of an integer data type. The “node” class can then be defined to include one or more data members that are of the “pointer” class data type. Consequently, when an object of

the "node" class is created, the "node" class object will contain one or more objects of the "pointer" class.

Accordingly, it is respectfully submitted that the Specification and Figures provide support for the elements "providing at least one child pointer within at least one of the plurality of nodes in the hierarchical node tree" and "storing a hint within the at least one child pointer, the hint being related to the corresponding child node, wherein the at least one child pointer further comprises a node slot number of the corresponding child node" recited in claim 1 (emphasis added).

Therefore, claim 1, and the claims that depend therefrom, satisfy the requirements under 35 U.S.C. § 112, first paragraph. Since claims 11 and 21 each recite elements similar to those of claim 1, claims 11 and 21, and the claims that depend therefrom, also satisfy the requirements under 35 U.S.C. § 112, first paragraph.

2. Claims 1, 11, and 21 Are Not Anticipated by Igata

Claim 1 recites a method for querying a structured document stored in its native format in a database, wherein the structured document comprises a plurality of nodes that form a hierarchical node tree. The method includes providing at least one child pointer within at least one of the plurality of nodes in the hierarchical node tree, wherein the at least one child pointer points to a corresponding child node in the hierarchical node tree, storing a hint within the at least one child pointer, the hint being related to the corresponding child node, wherein the at least one child pointer further comprises a node slot number of the corresponding child node, and utilizing the hint to determine whether to navigate to the corresponding child node during query evaluation.

Igata does not disclose, teach, or suggest the claimed subject matter.

Igata is directed to "a structured document search apparatus which has means to convert a query of structure and contents to a Boolean expression which has been used in a conventional full-text search engine, to thereby enable utilization of the high speed search performance of the full-text search engine" (col. 1, lns. 21-26 of Igata). Igata "provides a structured document search apparatus comprising: a hierarchical index which expresses the structure of each structured document such that the hierarchical relationship among document parts is expressed in a tree structure in which a meta part is treated as a single node; a text index in which is registered correspondence between each search key and a document identifier (document-ID) of a document which includes the search key, the search key including a character string in text data and a part identifier (part-ID) of a meta part; and search means which receives or inputs a user's query in a tree structure (hereinafter referred to as an "query tree") and refers to the hierarchical index and the text index in order to obtain a document corresponding to the query tree" (col. 4, lns. 36-49 of Igata).

(A)(i) Igata does not disclose, teach, or suggest "storing a hint within the at least one child pointer, the hint being related to the corresponding child node, wherein the at least one child pointer further comprises a node slot number of the corresponding child node"

Igata does not disclose, teach, or suggest "storing a hint within the at least one child pointer, the hint being related to the corresponding child node, wherein the at least one child pointer further comprises a node slot number of the corresponding child node," as recited in claim 1.

In the final Office action, the Examiner states:

> Applicant argues that the prior art fails to disclose; "storing a hint within the at least one child pointer the hint being related to the corresponding child node, wherein the at least one child pointer further comprises a node slot number of the corresponding child node".
>
> Examiner respectfully disagrees. Igata does disclose storing a hint within the at least one child pointer (Fig. 12C, item 41, PART1, PART2, PART3, Col. 14, lines 28-33, Igata), the hint being related to the corresponding child node (Fig. 12B, item 41, PART1, Col. 14, lines 16-20, Igata). Wherein the link in "PART ID" to PART 1, 2 corresponds to the hint claimed. The newly added limitation "within" was not defined in the specification. Therefore is [sic] has been rejected under 35 U.S.C. 112, first paragraph (See – 35 U.S.C. 112, first paragraph rejection discussed in this Office Action above). Examiner has interpreted the link in the link in "PART ID" to PART 1, and 2 as the hint "**within**" claimed.
>
> In addition, the newly added limitation including "a child pointer further comprises a node slot number" was not previously presented in the original claim language. However, Igata does disclose at least one child pointer further comprises a node slot number of the corresponding child node (Fig 2B, item: PART IDENTIFIER and CHILD LINK, Col. 7, lines 47-51, Igata).

(August 22, 2006 final Office action, pg. 14).

Item 41 in Figures 12B, 12C, and 13A of Igata, however, is a "work buffer" that is used to store the "names of document parts and links extending to the nodes" during registration of a document (col. 14, lns. 14-20 of Igata). Hence, the "links" from "work buffer 41" to the nodes of the hierarchical index, which are illustrated as arrows in Figures 12B, 12C, and 13A of Igata, cannot be construed as disclosing, teaching, or suggesting the "hint . . . related to the corresponding child node" recited in claim 1 because the nodes in the hierarchical index are not child nodes of "work buffer 41."

In addition, as shown in Figures 12B, 12C, and 13A of Igata, the "link" to the node "PART1 01" in Figure 12B disappears in Figure 13A and the "link" to the node "PART2 12" in Figure 12B disappears in Figure 12C. If the "link" can simply vanish, then Igata cannot possibly be relied upon to disclose, teach, or suggest "utilizing the hint to determine whether to navigate

to the corresponding child node during query evaluation," as recited in claim 1, because the "link" may disappear before the hierarchical index is completed, i.e., before any query is evaluated.

Further, the storage of "names of document parts and links extending to the nodes" in "work buffer 41" of Igata is temporary. Igata teaches that "the work buffer 41 is released" when the hierarchical index is completed (col. 14, lns. 48-50 of Igata). In other words, once the hierarchical index is completed, the "names of document parts and links extending to the nodes" stored in "work buffer 41" can be erased, written over, and so forth. Thus, if the "link" may no longer be stored in "work buffer 41" when a query is evaluated, then Igata cannot be relied upon to disclose, teach, or suggest "utilizing the hint to determine whether to navigate to the corresponding child node during query evaluation," as recited in claim 1.

Moreover, as discussed above, contrary to the Examiner's assertions, the Specification and Figures of the present invention provide sufficient support for "storing a hint within the at least one child pointer," as recited in claim 1 (emphasis added). As such, the Examiner cannot simply disregard the term "within" recited in claim 1. Given that the "work buffer 41" cannot be construed as disclosing the "child pointer" recited in claim 1, then storage of "names of document parts and links extending to the nodes" in "work buffer 41" of Igata cannot be construed as disclosing, teaching, or suggesting "storing a hint within the at least one child pointer," as recited in claim 1 (emphasis added).

With respect to the "PART IDENTIFIER" illustrated in Figure 2B of Igata, which the Examiner has construed as disclosing the "node slot number of the corresponding child node" recited in claim 1, the "PART IDENTIFIER" is an identifier for the element in the document

stored in the node. For example, the document in which the "hierarchical index" illustrated in Figure 2A of Igata is based upon the following:

Document 1
<Document>
<Part1> STRUCTURING </Part1>
<Part2> DOCUMENT </Part2>
<Part3> RETRIEVAL </Part3>
</Document>
Document 2
<Document>
<Part2> DOCUMENT </Part2>
<Part1> STRUCTURING </Part1>
<Part3> RETRIEVAL </Part3>
</Document>

As seen in Figure 2A, the node corresponding to "<Part1> STRUCTURING </Part1>" in "Document 1" has the "PART IDENTIFIER" value of "1" because it is the first element in the document. Whereas, the node corresponding to "<Part2> DOCUMENT </Part2>" in "Document 2" has the "PART IDENTIFIER" value of "4" because it is the fourth element in the document. Thus, the "PART IDENTIFIER" in Igata has nothing to do with a "child node" of the node in which the "PART IDENTIFIER" is stored. In fact, as clearly shown in Figure 2A of Igata, the node corresponding to "<Part1> STRUCTURING </Part1>" in "Document 1" and the node corresponding to "<Part2> DOCUMENT </Part2>" in "Document 2" do not even have child nodes. As such, the "PART IDENTIFIER" in Igata cannot be construed as disclosing the "node slot number of the corresponding child node" recited in claim 1.

Furthermore, claim 1 specifically recites that the "node slot number of the corresponding child node" is comprised in the "child pointer." Given that the Examiner is construing the "CHILD LINK" illustrated in Figure 2B of Igata as disclosing the "child pointer" recited in claim 1 and the fact that Igata does not disclose, teach, or suggest that the "CHILD LINK"

comprises the "PART IDENTIFIER," then Igata does not disclose, teach, or suggest "the at least one child pointer further comprises a node slot number of the corresponding child node," as recited in claim 1.

Accordingly, Igata does not disclose, teach, or suggest "storing a hint within the at least one child pointer, the hint being related to the corresponding child node, wherein the at least one child pointer further comprises a node slot number of the corresponding child node," as recited in claim 1.

(A)(ii) Igata does not disclose, teach, or suggest "utilizing the hint to determine whether to navigate to the corresponding child node during query evaluation"

Igata does not disclose, teach, or suggest "utilizing the hint to determine whether to navigate to the corresponding child node during query evaluation," as recited in claim 1.

In the final Office action, the Examiner states:

> Igata discloses . . . utilizing the hint to determine whether to navigate to the corresponding child node during query evaluation (Col. 21, lines 21-27, Igata).

(August 22, 2006 final Office action, pgs. 3-4).

The passage of Igata cited by the Examiner states:

> As described above, in the present invention, there are provided a hierarchical index which expresses the structure of each structured document, and a text index which is used for searching a document-ID on the basis of a search key composed of a part-ID and a character string in text data; a user's query received in the form of a tree structure is converted to a Boolean expression; and the text index is referred to in order to obtain a document-ID corresponding to the query tree. Thus, it becomes possible to search structured documents, while maintaining the high speed of conventional full-text search engines.

(Col. 21, lns. 21-31 of Igata).

As discussed above, the Examiner construes the "links" stored in "work buffer 41" illustrated in Figures 12B, 12C, and 13A as disclosing the "hint" recited in claim 1. The passage cited by the Examiner, however, does not disclose, teach, or suggest utilizing the "links" in "work buffer 41" during query evaluation.

Therefore, Igata does not disclose, teach, or suggest "utilizing the hint to determine whether to navigate to the corresponding child node during query evaluation," as recited in claim 1.

(A)(iii) The Examiner has not established anticipation under 35 U.S.C. § 102

Anticipation under 35 U.S.C. § 102 requires the disclosure in a single piece of prior art of each and every limitation of a claimed invention. (*See, e.g.*, Electro Med. Sys. S.A. v. Cooper Life Sciences, 34 F.3d 1048, 32 U.S.P.Q.2d 1017, 1019 (Fed. Cir. 1994)). The Examiner has failed to show that the elements discussed in sections (A)(i) and (A)(ii) above are disclosed in Igata.

Therefore, claim 1, and the claims that depend therefrom, are not anticipated by Igata. Given that claims 11 and 21 each recite elements similar to those of claim 1, claims 11 and 21, and the claims that depend therefrom, are not anticipated by Igata for at least the same reasons.

3. Claims 31-33 Are Not Anticipated by Igata

Claim 31, which depends from claim 1, recites wherein each of the plurality of nodes in the hierarchical node tree specifies a type of node, one or more nodes in the hierarchical node tree being of a text type and one or more other nodes in the hierarchical tree being of a non-text type.

Igata does not disclose, teach, or suggest the claimed subject matter.

Igata is directed to "a structured document search apparatus which has means to convert a query of structure and contents to a Boolean expression which has been used in a conventional full-text search engine, to thereby enable utilization of the high speed search performance of the full-text search engine" (col. 1, lns. 21-26 of Igata). Igata "provides a structured document search apparatus comprising: a hierarchical index which expresses the structure of each structured document such that the hierarchical relationship among document parts is expressed in a tree structure in which a meta part is treated as a single node; a text index in which is registered correspondence between each search key and a document identifier (document-ID) of a document which includes the search key, the search key including a character string in text data and a part identifier (part-ID) of a meta part; and search means which receives or inputs a user's query in a tree structure (hereinafter referred to as an "query tree") and refers to the hierarchical index and the text index in order to obtain a document corresponding to the query tree" (col. 4, lns. 36-49 of Igata).

(B)(i) Igata does not disclose, teach, or suggest "wherein each of the plurality of nodes in the hierarchical node tree specifies a type of node, one or more nodes in the hierarchical node tree being of a text type and one or more other nodes in the hierarchical tree being of a non-text type"

Igata does not disclose, teach, or suggest "wherein each of the plurality of nodes in the hierarchical node tree specifies a type of node, one or more nodes in the hierarchical node tree being of a text type and one or more other nodes in the hierarchical tree being of a non-text type," as recited in claim 31.

In the final Office action, the Examiner states:

> Applicant argues that the prior art fails to disclose; "each of the plurality of nodes in the hierarchical node tree specifies a type of node, one or more nodes in the hierarchical node tree being of a text-type and one or more other nodes in the hierarchical tree being of non-text type".
>
> Examiner respectfully disagrees. The argument relates to newly added limitations that were not previously disclosed by the original claim language. However, Igata does disclose the limitation where each of the plurality of nodes in the hierarchical node tree specifies a type of node (Col. 11-12, lines 24-28 and 7-9, a node type; respectively, Igata), one or more nodes in the hierarchical node tree being of a text-type (Col. 4, lines 24-31, text data of each document, Igata) and one or more other nodes in the hierarchical tree being of non-text type (Col. 4, lines 24-31 meta-part, Igata). Wherein the tree structure (Col. 4, lines 24-26, Igata) corresponds to the hierarchical node tree as claimed. To add, the tree structure of Igata includes both text nodes and non-text nodes (Col. 4, lines 26-29, in a tree structure in which "**meta part**" . . .; a text index in which **a character string contained in text data** of each document **is registered**, Igata). Therefore, Igata does not teach away from the claimed invention.

(August 22, 2006 final Office action, pg. 15).

As previously discussed above in sections (A)(i) and (A)(ii), the Examiner construes the "hierarchical index" illustrated in Figures 2A, 2B, 12B, 12C, 13A, and 13B of Igata as disclosing the "hierarchical node tree" recited in claim 1. With respect to claim 31, however, the Examiner now cites to the "Boolean logic tree" in Igata as disclosing "each of the plurality of nodes in the hierarchical node tree specifies a type of node" recited in claim 31. Appellant respectfully submits that the Examiner cannot simply construe the "hierarchical index" in Igata as disclosing the "hierarchical node tree" recited in claim 1, then construe the "Boolean logic tree" in Igata as disclosing the "hierarchical node tree" recited in claim 31, when the "hierarchical node tree" recited in claim 1 and the "hierarchical node tree" recited in claim 31 are one in the same.

In addition, as pointed out by the Examiner, Igata specifically teaches creating an "hierarchical index" to register the "meta parts" of a document, which is illustrated in Figures 2A, 2B, 12B, 12C, 13A, and 13B of Igata, and creating a separate "text index" to register the

"character strings" of the same document, which is illustrated in Figure 14 of Igata (see, col. 4, lns. 23-30). Claim 31, in contrast, only recites one (i.e., a single) "hierarchical node tree" that comprises both text nodes and non-text nodes.

Since the "hierarchical index" in Igata is separate and distinct from the "text index" and Igata does not disclose, teach, or suggest combining the "hierarchical index" and the "text index" into a single index, Igata does not disclose, teach, or suggest "wherein each of the plurality of nodes in the hierarchical node tree specifies a type of node, one or more nodes in the hierarchical node tree being of a text type and one or more other nodes in the hierarchical tree being of a non-text type," as recited in claim 31.

(B)(ii) The Examiner has not established anticipation under 35 U.S.C. § 102

Anticipation under 35 U.S.C. § 102 requires the disclosure in a single piece of prior art of each and every limitation of a claimed invention. (*See, e.g.*, Electro Med. Sys. S.A. v. Cooper Life Sciences, 34 F.3d 1048, 32 U.S.P.Q.2d 1017, 1019 (Fed. Cir. 1994)). The Examiner has failed to show that the element discussed in sections (B)(i) above is disclosed in Igata.

Therefore, claim 31 is not anticipated by Igata. Given that claims 32 and 33 each recite elements similar to those of claim 31, claims 32 and 33 are not anticipated by Igata for at least the same reasons.

4. Claims 34-36 Are Not Anticipated by Igata

Claim 34, which depends from claim 1, recites wherein the at least one node in the hierarchical node tree further includes at least one other child pointer, the at least one other child pointer pointing to itself or to an in-lined character array.

Igata does not disclose, teach, or suggest the claimed subject matter.

Igata is directed to "a structured document search apparatus which has means to convert a query of structure and contents to a Boolean expression which has been used in a conventional full-text search engine, to thereby enable utilization of the high speed search performance of the full-text search engine" (col. 1, lns. 21-26 of Igata). Igata "provides a structured document search apparatus comprising: a hierarchical index which expresses the structure of each structured document such that the hierarchical relationship among document parts is expressed in a tree structure in which a meta part is treated as a single node; a text index in which is registered correspondence between each search key and a document identifier (document-ID) of a

document which includes the search key, the search key including a character string in text data and a part identifier (part-ID) of a meta part; and search means which receives or inputs a user's query in a tree structure (hereinafter referred to as an "query tree") and refers to the hierarchical index and the text index in order to obtain a document corresponding to the query tree" (col. 4, lns. 36-49 of Igata).

(C)(i) Igata does not disclose, teach, or suggest "wherein the at least one node in the hierarchical node tree further includes at least one other child pointer, the at least one other child pointer pointing to itself or to an in-lined character array"

Igata does not disclose, teach, or suggest "wherein the at least one node in the hierarchical node tree further includes at least one other child pointer, the at least one other child pointer pointing to itself or to an in-lined character array," as recited in claim 34.

In the final Office action, the Examiner states:

> Applicant argues that the prior art fails to disclose; "one node in the hierarchical node tree further includes at least one other child pointer, the at least one other child pointer pointing to itself or to a [sic] in-lined character array".
>
> Examiner respectfully disagrees. The applied reference does disclose one node in the hierarchical node tree further includes at least one other child pointer (Fig. 12C, item: node PART1, Col. 14, lines 28-33, Igata), the at least one other child pointer pointing to itself or to a [sic] in-lined character array (Fig. 12C, node PART1, PART2, PART3, and PART2, Col. 14, lines 28-33, Igata). Wherein the pointer of the node including PART1 corresponds to the child pointer claimed; and the nodes including PART2, PART3, and PART2 correspond to the in-lined character array claimed.

(August 22, 2006 final Office action, pg. 16).

Appellant is at a loss as to how the Examiner can construe the "nodes" of the "hierarchical index" illustrated in Figure 12C of Igata as being an "in-lined character array" when the Examiner already construed those same "nodes" as disclosing the "nodes" recited in

claim 1. In addition, Igata does not disclose, teach, or suggest that any of the "nodes" of the "hierarchical index" illustrated in Figure 12C of Igata is an "in-lined character array." In fact, the term "array" is never mentioned in Igata.

Since node "PART2 12", node "PART3 23", and node "PART2 04" illustrated in Figure 12C of Igata cannot be construed as disclosing an "in-lined character array," the links (i.e., arrows) to each of those nodes cannot be construed as the "other child pointer pointing . . . to an in-lined character array," as recited in claim 34.

Further, none of the "links" (i.e., arrows) from node "PART1 01", node "PART2 12", node "PART3 23", and node "PART2 04" illustrated in Figure 12C of Igata, points back to the node in which the link originated. Hence, none of those "links" can be construed as disclosing the "other child pointer pointing to itself," as recited in claim 34.

Therefore, Igata does not disclose, teach, or suggest "wherein the at least one node in the hierarchical node tree further includes at least one other child pointer, the at least one other child pointer pointing to itself or to an in-lined character array," as recited in claim 34.

(C)(ii) The Examiner has not established anticipation under 35 U.S.C. § 102

Anticipation under 35 U.S.C. § 102 requires the disclosure in a single piece of prior art of each and every limitation of a claimed invention. (*See, e.g.*, Electro Med. Sys. S.A. v. Cooper Life Sciences, 34 F.3d 1048, 32 U.S.P.Q.2d 1017, 1019 (Fed. Cir. 1994)). The Examiner has failed to show that the element discussed in sections (C)(i) above is disclosed in Igata.

Therefore, claim 34 is not anticipated by Igata. Given that claims 35 and 36 each recite elements similar to those of claim 34, claims 35 and 36 are not anticipated by Igata for at least the same reasons.

5. Claims 1, 11, and 21 Are Patentable Over Igata in view of Manikutty

Claim 1 recites a method for querying a structured document stored in its native format in a database, wherein the structured document comprises a plurality of nodes that form a hierarchical node tree. The method includes providing at least one child pointer within at least one of the plurality of nodes in the hierarchical node tree, wherein the at least one child pointer points to a corresponding child node in the hierarchical node tree, storing a hint within the at least one child pointer, the hint being related to the corresponding child node, wherein the at least one child pointer further comprises a node slot number of the corresponding child node, and utilizing the hint to determine whether to navigate to the corresponding child node during query evaluation.

Igata and Manikutty do not, alone or in combination, disclose, teach, or suggest the claimed subject matter.

Igata is directed to "a structured document search apparatus which has means to convert a query of structure and contents to a Boolean expression which has been used in a conventional full-text search engine, to thereby enable utilization of the high speed search performance of the full-text search engine" (col. 1, lns. 21-26 of Igata). Igata "provides a structured document search apparatus comprising: a hierarchical index which expresses the structure of each structured document such that the hierarchical relationship among document parts is expressed in a tree structure in which a meta part is treated as a single node; a text index in which is registered

correspondence between each search key and a document identifier (document-ID) of a document which includes the search key, the search key including a character string in text data and a part identifier (part-ID) of a meta part; and search means which receives or inputs a user's query in a tree structure (hereinafter referred to as an "query tree") and refers to the hierarchical index and the text index in order to obtain a document corresponding to the query tree" (col. 4, lns. 36-49 of Igata).

Manikutty is directed to "[t]echniques . . . for changing data for an XML construct in a SQL/XML compliant DBMS" (col. 4, lns. 51-52 of Manikutty). In Manikutty, the "DBMS allows instances of XML type (also called XML instances, herein) to represent XML constructs and allows SQL constructs, such as rows, columns, tables, collections, LOBs, to store data for XML constructs" (col. 4, lns. 57-60 of Manikutty).

(D)(i) Manikutty does not cure the deficiencies of Igata

As discussed above in Sections (A)(i) and (A)(ii), Igata does not disclose, teach, or suggest "storing a hint within the at least one child pointer, the hint being related to the corresponding child node, wherein the at least one child pointer further comprises a node slot number of the corresponding child node" or "utilizing the hint to determine whether to navigate to the corresponding child node during query evaluation," as recited in claim 1.

Manikutty also fails to disclose, teach, or suggest "storing a hint within the at least one child pointer, the hint being related to the corresponding child node, wherein the at least one child pointer further comprises a node slot number of the corresponding child node" or "utilizing the hint to determine whether to navigate to the corresponding child node during query evaluation," as recited in claim 1.

Therefore, even if Manikutty were combined with Igata, the combination would neither teach nor suggest "storing a hint within the at least one child pointer, the hint being related to the corresponding child node, wherein the at least one child pointer further comprises a node slot number of the corresponding child node" or "utilizing the hint to determine whether to navigate to the corresponding child node during query evaluation," as recited in claim 1.

(D)(ii) The Examiner has not established a *prima facie* case of obviousness

To establish a *prima facie* case of obviousness, the Examiner must make three basic showings. First, there must be some suggestion or motivation, either in the references or in the prior knowledge generally available to one of ordinary skill in the art, to modify the reference or to combine reference teachings. Second, there must be a reasonable expectation of success. Finally, the prior art reference (or references when combined) must teach or suggest all the claim limitations. The teaching or suggestion to make the claimed combination and the reasonable expectation of success must both be found in the prior art, and not based on Applicant's disclosure. (*See, e.g.*, In re Vaeck, 947 F.2d 488, 20 USPQ2d 1438 (Fed. Cir. 1991)).

Since the Examiner has failed to make the three basic showings, no *prima facie* case of obviousness has been established. Therefore, claim 1, and the claims that depend therefrom, are patentable over Igata, in view of Manikutty. Given that claims 11 and 21 each recite elements similar to those of claim 1, claims 11 and 21, and the claims that depend therefrom, are patentable over Igata, in view of Manikutty for at least the same reasons.

6. Claims 1, 11, and 21 Are Patentable Over Igata in view of Bailey

Claim 1 recites a method for querying a structured document stored in its native format in a database, wherein the structured document comprises a plurality of nodes that form a hierarchical node tree. The method includes providing at least one child pointer within at least one of the plurality of nodes in the hierarchical node tree, wherein the at least one child pointer points to a corresponding child node in the hierarchical node tree, storing a hint within the at least one child pointer, the hint being related to the corresponding child node, wherein the at least one child pointer further comprises a node slot number of the corresponding child node, and utilizing the hint to determine whether to navigate to the corresponding child node during query evaluation.

Igata and Bailey do not, alone or in combination, disclose, teach, or suggest the claimed subject matter.

Igata is directed to "a structured document search apparatus which has means to convert a query of structure and contents to a Boolean expression which has been used in a conventional full-text search engine, to thereby enable utilization of the high speed search performance of the full-text search engine" (col. 1, lns. 21-26 of Igata). Igata "provides a structured document search apparatus comprising: a hierarchical index which expresses the structure of each structured document such that the hierarchical relationship among document parts is expressed in a tree structure in which a meta part is treated as a single node; a text index in which is registered correspondence between each search key and a document identifier (document-ID) of a document which includes the search key, the search key including a character string in text data and a part identifier (part-ID) of a meta part; and search means which receives or inputs a user's query in a tree structure (hereinafter referred to as an "query tree") and refers to the hierarchical

index and the text index in order to obtain a document corresponding to the query tree" (col. 4, lns. 36-49 of Igata).

Bailey is directed to "positional access of elements in a b-tree" (pg. 1, para. 0001 of Bailey). In Bailey, a "b-tree . . . is configured to store information that can be used to facilitate locating a value or data item at a specific ordinal position, or to perform other positional access operations. The b-tree is structured such that each index value in a non-leaf-level node is associated with a left count and a right count. These left and right counts indicate the number of leaf level values located in the sub-trees immediately to the left and right, respectively, of the index value. The left and right values are used to determine the path to take when descending the b-tree in search of a data item having a specific ordinal position" (pg. 1, para. 0004 of Bailey).

(E)(i) Bailey does not cure the deficiencies of Igata

As discussed above in Sections (A)(i) and (A)(ii), Igata does not disclose, teach, or suggest "storing a hint within the at least one child pointer, the hint being related to the corresponding child node, wherein the at least one child pointer further comprises a node slot number of the corresponding child node" or "utilizing the hint to determine whether to navigate to the corresponding child node during query evaluation," as recited in claim 1.

Bailey also fails to disclose, teach, or suggest "storing a hint within the at least one child pointer, the hint being related to the corresponding child node, wherein the at least one child pointer further comprises a node slot number of the corresponding child node" or "utilizing the hint to determine whether to navigate to the corresponding child node during query evaluation," as recited in claim 1.

Therefore, even if Bailey were combined with Igata, the combination would neither teach nor suggest "storing a hint within the at least one child pointer, the hint being related to the corresponding child node, wherein the at least one child pointer further comprises a node slot number of the corresponding child node" or "utilizing the hint to determine whether to navigate to the corresponding child node during query evaluation," as recited in claim 1.

(E)(ii) The Examiner has not established a *prima facie* case of obviousness

To establish a *prima facie* case of obviousness, the Examiner must make three basic showings. First, there must be some suggestion or motivation, either in the references or in the prior knowledge generally available to one of ordinary skill in the art, to modify the reference or to combine reference teachings. Second, there must be a reasonable expectation of success. Finally, the prior art reference (or references when combined) must teach or suggest all the claim limitations. The teaching or suggestion to make the claimed combination and the reasonable expectation of success must both be found in the prior art, and not based on Applicant's disclosure. (*See, e.g.*, In re Vaeck, 947 F.2d 488, 20 USPQ2d 1438 (Fed. Cir. 1991)).

Since the Examiner has failed to make the three basic showings, no *prima facie* case of obviousness has been established. Therefore, claim 1, and the claims that depend therefrom, are patentable over Igata, in view of Bailey. Given that claims 11 and 21 each recite elements similar to those of claim 1, claims 11 and 21, and the claims that depend therefrom, are patentable over Igata, in view of Bailey for at least the same reasons.

CONCLUSION

On the basis of the above remarks, Appellant respectfully submits that the final rejection should be reversed.

Respectfully submitted,
SAWYER LAW GROUP LLP

Dated: April 3, 2006

Erin C. Ming
Attorney for Appellant
Reg. No. 47,797
(650) 475-1449

APPENDIX OF CLAIMS

1. (Previously Presented) A method for querying a structured document stored in its native format in a database, wherein the structured document comprises a plurality of nodes that form a hierarchical node tree, the method comprising the steps of:

(a) providing at least one child pointer within at least one of the plurality of nodes in the hierarchical node tree, wherein the at least one child pointer points to a corresponding child node in the hierarchical node tree;

(b) storing a hint within the at least one child pointer, the hint being related to the corresponding child node, wherein the at least one child pointer further comprises a node slot number of the corresponding child node; and

(c) utilizing the hint to determine whether to navigate to the corresponding child node during query evaluation.

2. (Cancelled)

3. (Original) The method of claim 1, wherein the hint is a portion of the corresponding child node's name.

4. (Original) The method of claim 1, wherein utilizing step (c) further comprises:

(c1) receiving a query;

(c2) navigating to a current node of the plurality of nodes in the node tree associated with the structured document;

(c3) checking a hint stored in a first child pointer in the current node; and

(c4) navigating to the corresponding child node based on the checking in step (c3).

5. (Original) The method of claim 4, wherein checking step (c3) further comprises:

(c3i) comparing the hint to the query.

6. (Original) The method of claim 5, wherein navigating step (c4) further comprises:

(c4i) navigating to the corresponding child node if the hint matches the query; and

(c4ii) comparing the child node's name and namespace to the query to determine whether the child node satisfies the query.

7. (Original) The method of claim 5, wherein navigating step (c4) further comprises:

(c4i) skipping over the corresponding child node if the hint does not match the query.

8. (Original) The method of claim 4, wherein the utilizing step (c) further comprises:

(c5) determining whether another child pointer exists in the current node;

(c6) checking the hint stored in the another child pointer if the another child pointer exists, and navigating to the corresponding child node based on the checking;

(c7) repeating steps (c5) and (c6); and

(c8) navigating to a next node of the plurality of nodes in the node tree if the another child pointer does not exist, and repeating steps (c3) through (c7), wherein the next node becomes the current node.

9. (Original) The method of claim 1, wherein the structured document is written in Extensible Markup Language.

10. (Original) The method of claim 4, wherein the query is an Xpath or an Xquery expression.

11. (Previously Presented) A computer readable medium containing a computer program for querying a structured document stored in its native format in a database, wherein the structured document comprises a plurality of nodes that form a hierarchical node tree, the computer program comprising instructions for:

(a) providing at least one child pointer within at least one of the plurality of nodes in the hierarchical node tree, wherein the at least one child pointer points to a corresponding child node in the hierarchical node tree;

(b) storing a hint within the at least one child pointer, the hint being related to the corresponding child node, wherein the at least one child pointer further comprises a node slot number of the corresponding child node; and

(c) utilizing the hint to determine whether to navigate to the corresponding child node during query evaluation.

12. (Cancelled)

13. (Original) The computer readable medium of claim 11, wherein the hint is a portion of the corresponding child node's name.

14. (Original) The computer readable medium of claim 11, wherein utilizing instruction (c) further comprises:

(c1) receiving a query;

(c2) navigating to a current node of the plurality of nodes in the node tree associated with the structured document;

(c3) checking a hint stored in a first child pointer in the current node; and

(c4) navigating to the corresponding child node based on the checking in step (c3).

15. (Original) The computer readable medium of claim 14, wherein checking instruction (c3) further comprises:

(c3i) comparing the hint to the query.

16. (Original) The computer readable medium of claim 15, wherein navigating instruction (c4) further comprises:

(c4i) navigating to the corresponding child node if the hint matches the query; and

(c4ii) comparing the child node's name and namespace to the query to determine whether the child node satisfies the query.

17. (Original) The computer readable medium of claim 15, wherein navigating instruction (c4) further comprises:

(c4i) skipping over the corresponding child node if the hint does not match the query.

18. (Original) The computer readable medium of claim 14, wherein the utilizing instruction (c) further comprises:

(c5) determining whether another child pointer exists in the current node;

(c6) checking the hint stored in the another child pointer if the another child pointer exists, and navigating to the corresponding child node based on the checking;

(c7) repeating steps (c5) and (c6); and

(c8) navigating to a next node of the plurality of nodes in the node tree if the another child pointer does not exist, and repeating steps (c3) through (c7), wherein the next node becomes the current node.

19. (Original) The computer readable medium of claim 11, wherein the structured document is written in Extensible Markup Language.

20. (Original) The computer readable medium of claim 14, wherein the query is an Xpath or an Xquery expression.

21. (Previously Presented) A system for querying a structured document stored in its native format in a database, wherein the structured document comprises a plurality of nodes that form a hierarchical node tree, the system comprising:

a computer system coupled to at least one data storage device;

a database management system in the computer system; and

a storage mechanism in the database management system for providing at least one child pointer within at least one of the plurality of nodes in the hierarchical node tree, wherein the at

least one child pointer points to a corresponding child node in the hierarchical node tree, and storing a hint within the at least one child pointer, the hint being related to the corresponding child node, wherein the at least one child pointer further comprises a node slot number of the corresponding child node;

wherein the a database management system utilizes the hint to determine whether to navigate to the corresponding child node during query evaluation.

22. (Cancelled)

23. (Original) The system of claim 21, wherein the hint is a portion of the corresponding child node's name.

24. (Original) The system of claim 21, wherein the database management system is configured to receive a query, to navigate to a current node of the plurality of nodes in the node tree associated with the structured document, to check a hint stored in a first child pointer in the current node, to navigate to the corresponding child node based on the hint.

25. (Original) The system of claim 24, wherein the database management system is further configured to compare the hint to the query.

26. (Original) The system of claim 25, wherein the database management system is further configured to navigate to the corresponding child node if the hint matches the query, and to

compare the child node's name and namespace to the query to determine whether the child node satisfies the query.

27. (Original) The system of claim 25, wherein the database management system is further configured to skip over the corresponding child node if the hint does not match the query.

28. (Original) The system of claim 24, wherein the database management system is further configured to determine whether another child pointer exists in the current node, to check the hint stored in the another child pointer if the another child pointer exists, to navigate to the corresponding child node based on the hint, and to navigate to a next node of the plurality of nodes in the node tree if the another child pointer does not exist.

29. (Original) The system of claim 21, wherein the structured document is written in Extensible Markup Language.

30. (Original) The system of claim 24, wherein the query is an Xpath or an Xquery expression.

31. (Previously Presented) The method of claim 1, wherein each of the plurality of nodes in the hierarchical node tree specifies a type of node, one or more nodes in the hierarchical node tree being of a text-type and one or more other nodes in the hierarchical tree being of a non-text type.

32. (Previously Presented) The computer readable medium of claim 11, wherein each of the plurality of nodes in the hierarchical node tree specifies a type of node, one or more nodes in the hierarchical node tree being of a text-type and one or more other nodes in the hierarchical tree being of a non-text type.

33. (Previously Presented) The system of claim 21, wherein each of the plurality of nodes in the hierarchical node tree specifies a type of node, one or more nodes in the hierarchical node tree being of a text-type and one or more other nodes in the hierarchical tree being of a non-text type.

34. (Previously Presented) The method of claim 1, wherein the at least one node in the hierarchical node tree further includes at least one other child pointer, the at least one other child pointer pointing to itself or to an in-lined character array.

35. (Previously Presented) The computer readable medium of claim 11, wherein the at least one node in the hierarchical node tree further includes at least one other child pointer, the at least one other child pointer pointing to itself or to an in-lined character array.

36. (Previously Presented) The system of claim 21, wherein the at least one node in the hierarchical node tree further includes at least one other child pointer, the at least one other child pointer pointing to itself or to an in-lined character array.

EVIDENCE APPENDIX

None

RELATED PROCEEDINGS APPENDIX

None